INGENIEUR EN SCIENCES INFORMATIQUES ET

MASTER INFORMATIQUE FONDEMENTS ET INGENIERIE

RAPPORT DU PROJET WEB

Etudiant : Paul-Marie DJEKINNOU (SI5- Parcours AL)

Responsables : **Peter Sanders, François Michaudon, Hugo Mallet**

PLAN DU RAPPORT

Identifiant GitHub

PaulmarieDjekinnou

Tâches effectuées

- Récupération des données du back dans le front
- Implémentation des différents composants tels que ChartsFrance, ChartsRegions, ChartTauxIncidence et du RegionPicker

Stratégie employée pour la gestion des versions avec Git

Pour ce qui est de la gestion des versions, nous avons utilisé GitHub. Nous avions au départ un git pour la partie Backend de l'application. Par la suite, nous y avons rajouté la partie Frontend. Pour le développement en lui-même, nous avons utilisé un système de branches. Nous avions deux branches principales : **develop** et **main**. Au départ nous étions partis en développant des fonctionnalités du backend sur la branche **back**. Par suite de l'intégration de la partie Frontend, nous avons commencé à intégrer les fonctionnalités relatives au Frontend sur une branche **front.**

La branche **main** reste la branche principale et **develop** la branche pour intégrer les fonctionnalités avant de les pousser vers **main.** Pour certaines fonctionnalités, il a été nécessaire de les développer sur d'autres branches afin de pouvoir faire des tests sereinement. Par la suite, nous avons utilisé la branche **lab.** pour merger les différentes fonctionnalités des branches **back** et **front** pour nous assurer que **le tout était bien fonctionnel**. Et pour finir, nous avons poussé les fonctionnalités de cette branche vers la branche **main**.

Solutions choisies

Pour commencer pour l'implémentation de la récupération des données du back, nous avons créé un fichier qui regroupe toutes les fonctions de récupération des données de manière asynchrone. Nous avons choisi cette manière car cela permet d'encapsuler toutes les fonctions de fetching en un endroit et donc c'est plus facile d'utilisation ou de modification.

Pour ce qui est de l'implémentation de RegionPicker nous avons utilisé récupérer les régions pour les mettre dans les différentes options du menu déroulant et associer au clique une fonction qui permet d'effectuer une requête pour récupérer les données concernant l'option sélectionnée.Ce menin a été fait à partir de reactstrap et donc pour faire un menu déroulant il existe plusieurs alternatives.

Lorsque l'utlisateur choisi la région, nous avons une fonction qui va récupérer dans le backend, les diiférentes données hospitalières journalières liées à la région et une icône montre à l'utilisateur que cela est en cours de chargement. A la fin du chargement nous avons l'apparition de 3 graphes. L'ensemble des 3 graphes constituent le composant ChartsRegion. Ce composant récupère les données que lui passe le composant Home et va se charger de créer 3 graphes différents en

mappant pour chaque graphe la valeur que l'on veut pour ce graphe en particulier. C'est à dire que si la donnée est un objet de 3 clé valeur, alors on va mapper pour que chaque graphe aura la valeur d'une seule clé. Nous avons choisi de mettre 3 graphes différents car cela permettait de visualiser correctement et plus clairement les données. Une autre manière de faire aurait juste été de tout mettre dans un graphe avec une possibilité de visualiser toutes les valeurs en un endroit et donc l'utilisateur pourra plus facilement faire des comparaison entre les courbes.

Cette manière de faire a été utilisée pour faire les composants ChartsFrance et ChartTauxIncidence.

Il faut notifier que tous ces composants chart utilisent la librairie react-chartjs-2 mais il existe plusieurs librairie pour les graphes donc plusieurs alternatives sont possibles. Le composant (la page) Home se chargeait de faire les requêtes et donc de passer les données dont les différents autres composants charts ont besoin. Cela permet de centraliser l'endroit où s'effectue les requêtes et c'est aussi parce que tout était sur la même page donc nous n'avions pas un système de routes ce qui faisait que nous ne pouvions pas mettre les fonctions pour récupérer les données dans les composants.

Les fonctions récupération étaient très importantes dans le projet même si au départ nous pouvions utiliser des mocks. L'affichage des courbes sont l'une des fonctionnalités importantes pour le projet autant que les fonctions.

Nous avons beaucoup bloqué sur l'affichage des données car pour certaines requêtes notre backend répondait très vite mais parfois nous passions des données undefined ce qui faisait planter le code. Et au début les erreurs de linter étaient assez dures à gérer car nous ne comprenions pas très rapidement nos erreurs.

En temps de développement pour les fonctions de récupération nous avons mis 1h30 car tout d'abord on a travaillé avec les mocks puis les vraies requêtes et le linter. Ensuite pour chaque composant chart nous avons mis environ 2h30.

Le composant le plus optimal que nous avons développé est RegionPicker et moins optimal serait les composants chart car on aurait pu faire un composant général et le faire varier en fonction des données.